Exercices - Les variables (affectation et opérateurs)

Exercice 1

Quelles seront les valeurs des deux **variables** de type **Entiers X** et **Y** après l'exécution des **instructions** suivantes ?

X ← 10
Y ← X + 23
Fin

Ecrivez l'**algorithme** ci-dessus en langage C++ sous Geany puis compilez-le.

Exercice 2

Quelles seront les valeurs des deux variables de type **Entiers X**, **Y** et **Z** après l'exécution des instructions suivantes ?

X ← 7
Y ← 3
Z ← X + Y
X ← 2
Z ← Z - X
Fin

Ecrivez l'algorithme ci-dessus en langage C++ sous Geany puis compilez-le.

Exercice 3

Quelles seront les valeurs des deux variables de type **Entiers X** et **Y** après l'exécution des instructions suivantes ?

X ← 6
Y ← X + 3
X ← 2
Y ← Y - X
Fin

Ecrivez l'algorithme ci-dessus en langage C++ sous Geany puis compilez-le.

Correction de l'exercice 1

Pour calculer les valeurs des deux variables **X** et **Y**, nous devons suivre instruction par instruction notre **algorithme** pour retrouver le bon résultat :

a. **X ← 10** : X = 10 puisque cette instruction est une simple initialisation de la variable **X**.
b. **Y ← X + 23** : Signifie mathématiquement Y = X + 23. Or X = 10 d'après l'instruction précédente, ce qui nous donne une valeur Y égale à 10 + 23 soit 33.

Pour tester notre **programme**, nous pouvons écrire le code suivant pour effecter ce calcul et afficher les deux variables **X** et **Y** :

```
#include <iostream>
using namespace std;

int main() {
    int X = 10;
    int Y = X + 23;

    cout << "Valeur de X : " << X << endl;
    cout << "Valeur de Y : " << Y << endl;

    return 0;
}
```

Correction de l'exercice 2

Pour calculer les valeurs des variables **X**, **Y** et **Z**, nous devons calculer le résultat de chaque instruction de notre algorithme pour déterminer les différentes valeurs de chaque variable :

a. **X ← 7** : Soit donc X = 7. Aucun calcul ne nous est demandé dans cette instruction puisqu'il s'agit d'une simple affectation de la variable **X**.
b. **Y ← 3** : Soit Y = 3. Idem que l'instruction précédente, aucun calcul n'est demandé dans cette instruction.
c. **Z ← X + Y** : Pour cette fois-ci, nous allons calculer la somme des deux variables **X** et **Y** pour les mémoriser dans la variable **Z**. Ce qui nous donnes Z = 10 puisque X + Y = 7 + 3 = 10
d. **X ← 2** : De nouveau nous affectons une nouvelle valeur à la variable X soit donc X = 2.
e. **Z ← Z - X** : Après avoir calculé la valeur de **Z** dans l'étape 3 de notre algorithme, nous allons maintenant calculer la nouvelle valeur de la variable **Z** à partir des deux variables **Z** et **X** tel que : **Z = Z - X** soit donc Z = 10 - 2 = 8

Pour tester notre programme, nous pouvons écrire le code suivant pour ce calcul et afficher les deux variables **X** et **Y** :

```
#include <iostream>
using namespace std;

int main() {
    int X = 7;
    int Y = 3;    int Z = X + Y;    X = 2;    Z = Z - X;
    cout << "Valeur de X : " << X << endl;
    cout << "Valeur de Y : " << Y << endl;    cout << "Valeur de Z : " << Z << endl;
```

Correction de l'exercice 3

Pour calculer les valeurs des deux variables **X** et **Y**, nous devons suivre instruction par instruction notre algorithme pour retrouver le bon résultat :

a. **X ← 6** : Soit X = 6 puisque cette ligne est une simple affectation de la variable **X**.
b. **Y ← X + 3** : Signifie algorithmiquement Y = X + 3. Dans l'instruction précédente, nous avons affecté **X** à la valeur 6. Ce qui nous donne une valeur **Y** égale à 6 additionnée à la valeur de 3 soit 9.
c. **X ← 2** : Soit nous allons modifier la valeur de la variable **X** de l'ancienne valeur 6 à la nouvelle valeur 2 puisque cette ligne est à nouveau une simple affectation de la variable **X**.
d. **Y ← Y - X** : De nouveau, nous allons modifier la valeur de la variable **Y** en lui affectant la nouvelle valeur de **Y - X**, soit donc 9 - 2 = 7. D'où la valeur de la variable **Y** sera donc égale à 7

Pour tester notre programme, nous allons écrire le code suivant pour ce calcul et afficher les deux variables **X** et **Y** :

```
#include <iostream>
using namespace std;

int main() {
    int X = 6;
    int Y = X + 3;    X = 2;
    Y = Y - X;
    cout << "Valeur de X : " << X << endl;
    cout << "Valeur de Y : " << Y << endl;

    return 0;
}
```

Les 3 exercices codés sous C++ (et compilé sous MinGW)

Vous pouvez retrouver les différents fichiers sources des 3 exercices proposés et corrigés ci-dessous écrits en C++ sous l'éditeur **Geany**, compilés et testés à partir de notre espace **Github** :

- **https://github.com/ElectroRobot/les-variables-affectation-et-operateurs**